PCT/AU99/00873

#3 6-11-01

09/807338

REC'D 08 DEC 1999

WIPO PCT

AUSTRALIA

Patent Office
Canberra

I, KAY WARD, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. PP 6443 for a patent by CiTR PTY LTD filed on 12 October 1998.

COMMONWEALTH OF AUSTRALIA
AUSTRALIA
PATENT OFFICE

WITNESS my hand this
Twenty-sixth day of November 1999

K. Ward

KAY WARD
TEAM LEADER EXAMINATION
SUPPORT AND SALES

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

P/00/009
Regulation 3.2

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Invention Title: "MANAGEMENT OF PATH SELECTION IN A COMMUNICATIONS NETWORK"

The invention is described in the following statement:

TITLE

MANAGEMENT OF PATH SELECTION IN A COMMUNICATIONS NETWORK

FIELD OF THE INVENTION

This invention relates to the management of connections in a large scale heterogeneous communications network, such as those operated by telecommunications utility companies and utilised by different carriers and service providers. In particular the invention relates to a method and apparatus for selecting paths in a broadband network that may be provided to customers requiring a communications service.

BACKGROUND TO THE INVENTION

The term "communications network" as used in the specification, is meant to encompass networks suitable for voice telephony and for data communications. Such communications networks may be suitable for switching and transporting voice, data, sound and/or image traffic, otherwise referred to as broadband or "multimedia" communications.

Existing communications networks are characterised by a number of transmission mediums using a variety of network technologies, protocols, software applications and equipment sourced from different vendors. Whilst much of the equipment includes management functions, such as monitoring, test and alarm features, the centralising, handling and controlling of network management functions in a complex multi-vendor environment is a significant problem.

A further problem in a heterogeneous network - which might include customer access technologies (ADSL, HFC), core network technologies (ATM, frame relay) and transmission technologies (SONET/SDH, WDM) - is that the management of end-to-end connections is typically conducted according to a lowest common denominator philosophy. The services provided by the network are limited to those able to be supported by the least capable equipment in the network. This philosophy is very ineffective

in utilising the full capability of the diverse communications paths available in a network to meet particular service requirements of customers.

Glossary

AAD:	ATM access device
ADSL:	advanced digital subscriber line
ATM:	asynchronous transfer mode
CMIP:	common management information protocol
CORBA:	common object request broker architecture
EMS:	element management system
HFC:	hybrid fibre-optic co-axial
NMS:	network management system
NTU:	network terminal unit
OSS:	operational support system
VPC:	virtual path connection
SDH:	synchronous digital hierarchy
SNMP:	simple network management protocol
SONET:	synchronous optical network
TCP/IP:	transmission control protocol / Internet protocol
TL/1:	Bellcore interface protocol for network management
VCI:	virtual circuit identifier
VPI:	virtual path identifier
WDM:	wave division multiplexing

OBJECT OF THE INVENTION

It is an object of the present invention to provide a connection manager for selecting paths from a plurality of paths available in a communications network to route broadband traffic between predetermined locations in the network which ameliorates or overcomes at least some of the problems associated with the prior art.

It is another object of the invention to provide a path selection method for use in a communications network contributing to cost effective use of path features for routing broadband traffic between predetermined locations in the network.

Further objects will be evident from the following description.

DISCLOSURE OF THE INVENTION

In one form, although it need not be the only or indeed the broadest form, the invention resides in a connection manager for selecting paths from a plurality of paths available from service providers in a communications network to route broadband traffic in the network, wherein the connection manager includes:

(a) a connection model whereby the service provider indicates functional features supported by each path in the network and locations of terminations for respective paths;

(b) a cost model associated with the connection model that exposes to clients the cost of using the functional features for each path; and

(c) client processing means, operated in response to a requirement for a connection with desired features between two locations in the network,

    (i) to identify from the connection model in light of the desired features, suitable candidate paths for routing communications traffic between the two locations and

    (ii) to determine from the candidate paths and on the basis of cost exposed by the cost model, an optimal selection of paths connecting said locations.

Preferably the functional features indicated by the connection model include one or more of the following categories:

(i) communications protocol;

(ii) transmission rate;

(iii) availability of the path;

(iv) average error rate; and/or

(v) fault reporting requirements.

Suitably, individual terminations at the same location that have common attributes are represented as termination groups.

Preferably the cost exposed by the cost model reflects the resources required to implement a path having a particular set of features.

The path cost may be determined in accordance with the service provider's business rules or technical requirements, including one or more of:

(i) number of network elements involved in the path;

(ii) reduction in network capacity experienced in implementing the path; and/or

(iii) funds required to implement the path.

In one form the cost model represents path cost as a data structure which is interpreted by the processing means.

Suitably the data structure comprises a graph of cost nodes wherein each node specifies the cost of particular features or sets of features for respective paths.

The cost nodes in the graph may be either internal for representing links between internal terminations in the connection model or external for the terminations at said predetermined locations.

In another form the cost model represents path cost as code which is executed by the processing means.

Suitably the processing means for executing the code is an implementation of a Turing machine.

If required the cost model further exposes the delay in implementing functional features supported by the path.

Where the cost model indicates implementation delay, the client requirement for a connection may include a desired minimum delay.

In another form the invention resides in a selection method for selecting paths from a plurality of paths available from a service provider in a communications network to route broadband traffic in the network, including the steps of:

(a) the service provider creating -

(i) a connection model that indicates functional features supported by each path in the network and locations of terminations for respective paths and

(ii) a cost model associated with the connection model that exposes to clients the cost of using the functional features for each path; whereby

(b) a client, requiring a connection with desired features between two locations in the network -

(i) identifies, from the connection model in light of the desired features, suitable candidate paths for routing communications traffic between the two locations and

(ii) determines, on the basis of the cost exposed by the service provider, an optimal selection of paths connecting said locations from the candidate paths.

Suitably the connection model is created to reflect the network elements deployed by the service provider.

The connection model can also reflect the business and engineering policies adopted by the service provider.

BRIEF DETAILS OF THE DRAWINGS

To assist in understanding the invention preferred embodiments will now be described with reference to the following figures in which:

FIG. 1 is a diagram of a heterogeneous communications network including a hierarchy of connection managers;

FIG. 2 is a diagram illustrating the structure of a connection manager of a first embodiment;

FIG. 3 is a diagram of a world view from the perspective of the abstract connection model of the first embodiment;

FIG. 4 is an illustration of the physical architecture of an example network;

FIG. 5 is a top level view of a connection model for the example network of FIG. 4;

FIG. 6A is a schematic diagram of an access device;

FIG. 6B is a cost graph for the access device of FIG. 6A;

FIG. 7A is a schematic diagram of a multiplexer;

FIG. 7B is a cost graph for the multiplexer of FIG. 7A;

FIG. 8A is a schematic diagram of an edge switch;

FIG. 8B is a cost graph of the edge switch of FIG. 8A;

FIG. 9 is a cost graph of a core switch supporting local switching;

FIG. 10 is an alternative cost graph of the core switch of FIG. 9;

FIG. 11 is shows a proposed (cost-free) link between two cost graphs;

FIG. 12 is shows a link between the cost graphs of FIG. 11;

FIG. 13A shows a proposed link between two further cost graphs;

FIG. 13B shows an aggregated cost graph of the two further cost graphs;

FIG. 14 shows a aggregated cost graph for the core domain of the network of FIG. 4;

FIG. 15 shows a cost graph used for modeling the network of FIG. 4;

FIG. 16 shows the cost graph of FIG. 15 re-shaped to illustrate a preferred path selection method;

FIG. 17 illustrates the principles of the path selection method; and

FIG. 18 illustrates the path selection method applied to the re-shaped cost graph of FIG. 16.

DETAILED DESCRIPTION OF THE DRAWINGS

The embodiment of the invention is described in the environment of a heterogeneous communications network 10 as illustrated in FIG. 1. The connection manager of the embodiment participates in the service activation and service assurance processes of large communications networks. The connection manager is suited to use in relation to broadband communications products which have significant complexity at the "Network Layer" (as defined by the ITU-T layered management model) - such as ATM, SDH, IP and bundled broadband products. The connection manager supports fault, configuration and security activities at the Network Layer and can cooperate with other systems performing these functions for subsets of the communications network. The connection manager of the embodiment

resides in a network management layer 30 between the service layer 20 and the network element layer 40.

The service layer 20 typically includes service order systems 21 which institute the creation of new connections and facilitates the query, modification and deletion of existing connections, service assurance systems 22 which facilitate the test and repair of existing connections and pre-sales systems 23 which support pre-sales activities including inquiries regarding available connection characteristics, connection cost and time frame. Examples of service layer systems include service order, customer network management (CNM), test and repair or wholesale gateway.

The network element layer 40 typically includes the hardware for providing network services such as switching or transmission, for example ADSL/HFC customer access technologies 41, ATM core network broadband technologies 42, and transport technologies 43 such as SONET/SDH or WDM. The network element hardware may be conceptually considered to reside in different "domains" and is typically also proprietary in nature. Accordingly, the network element hardware generally uses proprietary or compatible network element managers which act as proxies for many of the network elements.

Examples of network element managers are EMS systems 44 and 45 for the ADSL/HFC hardware, the NMS 46 for the ATM core hardware and the vendor specific NMS 47, 48 for the transport domain. Although the network element managers manage many network elements, they expose each network element as an individual entity. Thus in other embodiments, the connection manager may interface directly to the network elements.

The network management layer 30 of the embodiment illustrates the flexibility of the connection manager. A first connection manager 31 is interfaced to the EMS systems 44 and 45 for managing the customer access domain 40A. The functional flexibility of the connection manager arises from its ability to manage the functionally different requirements of the switch matrix EMS 44 and the AAD EMS 45. A second connection manager 32 manages the core domain 40C and a third connection manager 33 is interfaced to the vendor NMS systems 47 and 48 in the transport

domain 40T. The transport domain illustrates the ability of the connection manager to handle disparate vendor equipment. The connection managers include interfaces which communicate using the CMIP, SNMP, TL/1 or proprietary protocols as required. These interfaces may be adapted to suit particular vendors' equipment, current or future.

A fourth connection manager 34 is interfaced with the three domain connection managers 31, 32 and 33 for the purpose of cross-domain connection management. The cross-domain manager 34 level accepts end-to-end connection instructions for the entire network, it determines which paths through the underlying networks are available and issues connection instructions to the domain connection managers as appropriate. The connection task is thus delegated to the appropriate domain connection managers. Any network events - including faults, congestion and usage - that effect connections, are reported back to the service layer enabling immediate identification of the client or customer and the service being provided.

Although shown as four separate managers, the network management layer 30 may be viewed as undertaking the overall connection management function for the network, with the cross-domain connection and domain connection being managed at different levels. Thus the network wide connection requirement is simplified step by step so that each level of connection management can be optimised to manage the portions of the network under its control. However, the separate connection managers illustrate the distributed nature of a network wide connection manager 35 which may be geographically distributed across a large number of sites and network operations centres.

Network Models

The connection manager provides flexible network modeling tools for representing a service provider or network owner's view of broadband connections. The key concepts for these representations are:

(i) "paths" which represent the owner's view of a connection, such as ATM PVC;

(ii) "terminations" where the path is manifest outside of a network, such as an ATM VPI, VCI and cable, or a customer NTU; and

(iii) "features" which are the external selectable characteristics of the path visible at its terminations, such as quality of service, bit rate or path diversity.

Conceptually a path can negotiate many network elements and protocols, such as end-to-end SDH connections implemented using SDH switches and WDM transmission.

<u>Connection Manager Structure</u>

The structure of the connection manager 35 of the embodiment is described in relation to FIG. 2, as it might be deployed in relation to a particular network. A connection model 36 is used to expose to clients the network and its services, which model may be implemented by the core software 37. Network adapters 38 are provided to interface with network elements, EMS or other NMS. Service adaptors 39 are provided to interface to existing service and test OSS, whilst peer OSS adaptors 50 interface fault, security, accounting and like support systems.

The connection manager 35 supports several fundamental operations relating to the life cycle of a path. The service provider or network owner may instruct that a path be *reserved*, *created* or *changed*, which results in the automatic selection, allocation and configuration of appropriate network equipment to implement a connection with the specified features between specified terminations. An *override* operation gives the owner the ability to influence the selection, whilst a *remove* operation frees the allocated network equipment.

The connection manager allows the determination of which features are supported, in what combinations and at what localities in the network. Terminations and paths may be searched and listed, and the termination which best supports a given set of features in a locality suggested to a client by the connection manager. A *preview* operation facilitates - without actual performance - prediction of the validity of a life cycle operation, the amount of resource that the path will require, and the time required to provision the connection.

In relation to network assurance, the connection manager includes *test* operations to automatically verify the implementation or operation of a connection, invoking network test hardware as necessary. The *repair* operations rectify faults, either internally or by delegating to some other OSS or operator. The connection manager performs alarm derivation and correlation, by taking equipment oriented alarms and producing path oriented alarms. The correlation can reflect the network topology, fault severity and network operational rules.

The connection manager 35 of the embodiment preferably uses a CORBA IIOP architecture to interface to both the service layer and the network layer. The service layer interface and the network model can be adapted to present some standard data models, such as ETSI 600-653 or ATM Forum M4, or adapted to existing service layer interfaces. All connection manager objects can be annotated with the names and identifiers required by external systems, for example customer circuit identifiers. A shell adaptor is also provided to minimise the skill and work needed to interface with an SNMP, TLI or CMIS system on the network side.

The core software 37 of the connection manager also allows for plug-ins 51 in order to provide for coded replacements to overcome specific limitations to core algorithms.

The connection manager is a high-availability system supporting on-line changes to configuration with back-out, on-line database backup, replicated databases and redundant hardware. Depending on configuration, the connection manager will support 10,000 transactions per hour on one mid-range server machine (for example Hewlett Packard J-class). This typically corresponds to a network with 50 million installed paths with a typical operational latency is 0.3 seconds. The connection manager will typically process 10 network alarms per second on one mid-range server. This typically corresponds to 1 million to 5 million installed paths. A low level event correlation system can be provided to smooth out alarm storms.

One connection manager installation can be distributed, as indicated above, over a number of server machines. A distributed installation on up to 10 machines would be typical, as transaction and alarm processing scale approximately linearly over this range. Such installation can suitably support HP UX or Solaris on SUN Solaris, Microsoft's NT on Intel or PA-RISC operating systems. Oracle database and Orbix ORB are also used in the preferred embodiment.

FIG. 3 shows a view of the world from the perspective of a connection model, considered in the abstract. The connection model is a framework for describing communications systems involving connections. In particular, the abstract connection model 52 of the embodiment is a distributed, object oriented way of representing the state and operations required to manage the network layer 30 of a broadband communications network. The service layer 20 is effectively the driver for the connection model in that providing function to the service layer is the role of the connection model.

In order to address requests from the service layer the connection model delegates to either the network element layer 40 - in the form of either network elements 53 or network element managers 54 or other providers at the network layer - for example other connection managers 55 or network service provider (NSP) 56. The choices involved in performing delegation include: (a) to which subordinates are functions delegated? (b) how are super-functions mapped to subordinates? (c) what is the sequencing of subordinate operations? (d) what actions occur when a subordinate operation fails? and (e) how are changes in the network configuration and network engineering policies reflected?

The abstract connection model 52 must be instantiated for operational use, with the instantiation depending on:

(i) the particular networking technology employed by the network owner;

(ii) the network owner's engineering rules; and

(iii) the network owner's service level requirements.

It is the model's instantiation 36 that gives meaning to the components of the model, such as path, feature and termination. When a connection model is instantiated, each of the abstract concepts that it presents will have a precise meaning. That meaning is conveyed by the model's descriptive text, rather than the syntax or semantics of the model's interfaces. Furthermore, a model instantiation will have objects instantiated against it which objects will conform to both the abstract connection model and the instantiated model. This development process may be contrasted with a traditional approach, wherein objects are instantiated against one fixed model.

The development of a connection manager application normally includes the following three stages:

- Network analysis and design – the focus of this stage is to define the architecture of the network to be managed and to analyse the characteristics of each component of the network.
- Connection manager installation – the focus of this stage is to use the mechanisms supported by the core software to specify the rules for how the network should be managed. The installation is the outcome of this stage.
- Run time – once a connection management system is installed, paths can be created through the network to provide communications services.

Basic Concepts

The basic concepts for connection management used by the connection model are the path-termination-feature concepts introduced briefly above. A *feature* is a characteristic of service manifest to a client or customer, for example: ATM protocol, 64kb/s data rate and unavailability for less than 1 minute/year. Features are installed on connections, which often requires feature values. The feature Maximum Bit Rate has a value specifying what the maximum bit rate is, for example Maximum Bit Rate = 256 kb/s. A feature with values applied to a connection is referred to as an *installed feature*.

A *path* is provided by a network and is fully characterised by the installed features of the path (the *path features*), a set of *terminations*

exposed to the client and a set of installed features for each termination (the *termination features*). A *network* represents the ability to manage paths and is used to create new paths and list existing paths. Paths are always totally contained within exactly one network.

Networks manifest themselves in terminations, and terminations are contained within one network. A termination can participate in a finite number of paths, typically one, but potentially more. A path in a network will have one or more (usually two) terminations. Single termination paths may represent loop-back and multiple terminations may represent multiple drop (for example CSMA) or a closed user group (for example a voice private network). Paths may share terminations, this may express a multi-serving capability (such as the set of customers using a billing server).

Connection Manager

The connection manager 35 presents an architecture for assembling a working network management system. The core software 37 provides an abstract connection model that can be configured to reflect the characteristics of the particular network equipment deployed by the network owner. The operative connection model 36 can also reflect the business and engineering policies of the network owner, in other words, the knowledge that a human operator would apply if they were performing the connection manager functions manually. The core software 37 assumes that the interface to the network supports the connection model 36, preferably expressed in CORBA.

The network adaptors 38 are constructed from shells, which include code fragments, applications program interfaces and libraries, to integrate with any contemporary network management interface. The shell adaptors are designed to inter-work with stack products, such as those provided by Vertel or Hewlett-Packard, in order to provide simple interfaces into complex protocols such as CMIS or TL/1.

The service adaptors 39 provide an interface between the network's service management layer OSS. Existing operation support systems generally have a proprietary interface, although there are some emerging standards including the US Federal Communications Commission's

"Gateway". Printed paper or a character terminal are common interfaces. The deployed connection manager 35 preferably has an adaptor to automate the interface between the service management layer and the core software 37.

The peer OSS adaptors 50 may be provided to allow the connection manager to inter-work with other network support systems in the network management layer, such as the alarm and security OSS. The core software 37 has certain in-built algorithms that determine its behaviour. These algorithms are configurable, allowing the network owner or systems integrator to tailor the core software for their specific needs. However, there are limitations to the amount of flexibility that can be achieved through configuration. Accordingly the connection manager 35 allows for use of plug-ins 51, which are coded replacements for selected core algorithms.

Distributed Object Model

As the connection manager is a network layer manager, it is only concerned with modeling network-level concepts. The first network level concept is "connection". The connection model 36 of the embodiment is a distributed object model, preferably expressed in CORBA interface definition language (IDL). In accordance with the concepts introduced earlier, there are three types of objects in the model, namely:

(i) path objects that represent connections;

(ii) termination objects that represent where the connections are physically manifest; and

(iii) network objects which are the fabric that can create connections.

Network Objects

The network object is a container of path objects and termination objects. Network objects form a hierarchy, where some network objects are superior to others. Network objects will typically form a strict containment hierarchy, though the connection manager allows any non-cyclic structure. Network objects can represent: individual network element instances, groups of network elements organised by some owner determined criteria, such as geographic domains or functional domains; sub-networks that are

managed by some other NMS, such as a vendor NMS; cross-domain networks that aggregate several domain network objects, such as those identified 40A, 40C and 40T in FIG. 1.

Network objects support the following operations: listing the capabilities of the network object; listing the characteristics of the paths that the network objects can create; creating paths having specified terminations and features; previewing path creation; searching for paths, terminations and sub-networks having specified characteristics. Network objects may be configured as follows: assigning identity, description, meaning; defining the relationships between the network objects (for example, a containment tree structure); defining the connections between subordinate network objects; and the characteristics of the paths they can create. The connection manager provides no specific operations for creating network objects, establishing relationships between network objects or creating connections between subordinate network objects.

<u>Path Objects</u>

Path objects represent the connections formed by network objects. They correspond to some real-world connection concept. This could be for example:

(i) a physical connection, such as a bearer distribution frame;

(ii) a switched connection, such as an ATM virtual circuit; or

(iii) some abstract relationship, such as the relationship between a customer and their Internet service provider (ISP).

A path object is always contained within one network object. When network objects form a hierarchy, a network object may implement that path by delegating portions of the implementation to sub-paths in its subordinate networks. Paths are characterised by terminations and features. Terminations describe where the path is manifest, features describe externally visible characteristics. A path generally has two terminations.

A feature has a name and optionally a value. The value have can have arbitrarily complex structure. A feature may be able to take one of a finite number of values. Features are applied to either the path itself, or terminations on the path. This permits termination-specific features to be

modelled - as required for asymmetrical paths. It is common for features to interact - that is the existence of any one feature affects the ability of a path to support another feature. The core software supports a feature interaction connection model that allows the modelling of optional, compulsory, mutually exclusive and invalid combinations of features.

Paths support life-cycle type operations. This allows for several levels of completeness of the path's implementation. The typical levels of implementation of paths are:

(a) *design* - The path consumes no resources, other than those minimally needed to record its characteristics. Design state paths need not obey any rules.

(b) *reserve* - The path is fully implemented, except the last step which would enable service.

(c) *installed* - The path is implemented into the equipment to enable service.

(d) *deleted* - The path no longer exists, but the memory of it is kept for audit purposes.

Paths have a cost, which represents the amount of resource required to implement this path. The cost allows a client to rationally choose between several candidate paths, each of which is capable of supporting their needs. Path objects support the following operations: deletion; changing the features, terminations or implementation completeness; preview operations for the above; and listing the path attributes. Path objects themselves do not have any configuration. However the containing network objects may be configured for the following path-related functions: the list of features that could apply to a path in the network along with the feature interaction and locality rules.

<u>Termination Objects</u>

Termination objects represent where path objects are (or may be) manifest. They correspond to some real-world concept for example: a physical termination, such as a cable; one channel multiplexed over some bearer such as an ATM virtual circuit or SDH container; a grouping of multiplexed channels, such as an ATM virtual path. A termination object

within one network object. A network may express an effectively infinite number of terminations, for example an ATM network may model each VPI/VCI as a termination. Even coarser grained modelling than the ATM example will have large numbers of terminations. To allow these to be easily managed, the connection manager supports the grouping of terminations into "termination groups".

The termination groups of one network object preferably form a containment tree. The upper layers of the tree are generally abstract groupings, typically based on physical location, such as city or central office building. The lower layers of the tree are typically more concrete, such as interface card or cable. The leaves of the tree are the lowest level of modelled termination, such as cable, virtual channel, or the like. The connection manager makes no clear distinction between termination and termination group - termination groups appear higher in the tree and pure terminations occur toward the leaves. The network object's state what levels in the tree it is prepared to establish connections between - typically this may only be the leaf nodes. The meaning of terminations, and the structure of termination groups is specified during configuration of the core software.

Each termination exposes a cost for supporting paths with particular features. This allows clients to make a rational choice between several possible terminations, each of which could meet their needs. The meaning of cost is specified by the core software configuration. The cost values may also be specified by the core, however it would be common for a deployment to reflect values from the network equipment, such as congestion (via the equipment adaptors).

Termination objects support the following operations: describe the termination; describe the termination group structure (actually an operation on the containing network); navigate through the termination group structure; find, within a termination group, the lowest cost free termination capable of supporting a particular set of features. As there are a huge number of terminations in a typical network, most deployment will configure terminations at the level of a termination group. The characteristics of the

individual terminations will be derived from the network equipment adaptors. Termination group objects support the following configuration: the termination group structure and (optionally) the cost of members of the group.

Concepts of Cost

When the core software is implementing a path, or changing an existing path, it may have several alternate methods. Each alternate will require a certain amount of the network owner's equipment resource. For example, bandwidth on a optical fibre, dedicated use of a port card, or share of switch capacity. The core suitably implements a path using the alternative that requires the least resource. To allow the connection manager to determine least resource, the core software uses the concept of "cost". Each candidate path has a cost and each candidate termination has a cost. The connection manager suitably makes the simple choice of 'least cost'. The power comes from the meaning assigned to, and the method of calculating the cost.

Network objects close to the service layer typically have a huge number of candidate paths. One approach that such an object could use, is to execute the preview-path-creation operation on for each of the candidate paths, then choose the one that has the lowest overall cost. This direct approach is practically infeasible when, as is typical, there are multi-millions of candidate paths. To bypass this practical difficulty, the connection manager implements the concept of cost modelling. A cost model is a way for a network's client to efficiently predict the cost of paths. This allows the client to check the millions of options, without doing millions of requests.

Cost Model

A cost model preferably predicts the cost of paths based on termination groups, rather than individual terminations. It supports feature-dependent costs. Cost models may be arbitrarily complex and precise. For example, a highly precise model would specify the cost for paths between each termination group pair. A coarse model would specify a single cost for all paths. Intermediate models that exhibit a arbitrary

mixture of termination dependent cost and fixed cost may also be supported. Although there is a cost concept for terminations, there is no concept of cost model. This is because termination selection is not subject to the combinatorial explosion problem that haunts path selection.

A cost offer is a named cost model that applies for a specified period of time. A cost offer is the mechanism by which a network exposes the cost model for its paths. As a cost offer includes a validity time, clients can restrict the number of enquiries for cost model that they make. The present embodiment of the connection manager supports validity times from the order of seconds upwards - shorter times need higher computing resource.

There are two options available for determining costs for a network object, namely:

(i) *fixed cost* - where a configured cost is returned; and

(ii) *mapped cost* - where a value for cost is returned that is derived from the costs of its subordinate networks.

The mapped cost option converts the units of cost in each subordinate network into the units of cost for the present network. These two models apply for termination cost, path cost, and path cost offers.

The mapped cost offer is a particularly powerful mechanism, as it allows a network to express a very precise and up-to-date cost model (that is, one that reflects details of its subordinate networks) for zero configuration cost. In general, mapped cost is a very effective way to transferring rational decision making capability from subordinate networks to superior networks. This is necessary because the subordinate networks, close to the equipment, understand the equipment-related intricacies, while the superior networks, close to the service layer, have a sufficiently broad view to perform network-wide optimum resource allocation.

The cost model of the embodiment, which uses a data structure in the form of a traversable graph of cost nodes, includes three major aspects. Each aspect is designed to solve cost-related problems in the different stages of the management application development. The aspects of a preferred modelling process are:

- Cost graph creation - A cost graph notation is defined. This notation can be used during the design stage to assist system integrators and network engineers to analyse the cost model at different network levels.
- Cost model specification - During specification stage, the core software can be used to translate the cost graph representation of the network cost model to the format that can be loaded into a connection manager system.
- Route selection algorithm - Based on the internal representation of the cost model, the route selection algorithm and the cost-based routing algorithm are used for path creation.

Here the term route means a set of sub-paths that together implement a path between terminations at two selected locations in the network.

Cost Graph

A cost graph is a graphical representation of a cost model. In some cases, a cost model can be represented by a single cost graph; in some other cases, a number of disconnected cost graphs are needed to represent a single cost model. In order to understand the cost model and how it may be used to assist the selection of a path route, an example network is used. The physical architecture of the example network is illustrated in FIG.4.

The physical network contains a number of access devices, such as multiplexers, marked as A1 to A4; a number of edge switches marked as E1 to E3 and two core switches, C1 and C2. These physical components form two logic groups: an access domain, which provides the customer access front end to the network, and a core domain, which provides the communication back bone of the network. Each access device has a number of customer termination points, such as A to H, and is linked to an edge switch. An access device cannot switch, that is, no path can be created between two terminations connected to the same access device without going out to an edge switch. In the above example network, all possible paths contain one of the following sub-paths: A_i-E_j-A_k, A_i-E_j-E_k-A_{ln}, A_i-E_j-C_k-E_l-A_m, A_i-E_j-C_k-C_l-E_m-A_n. (i, j, k, l, m, n = 1 to 4). Using the connection

model discussed earlier, the network can be modelled as a cross-domain network containing two domain sub-networks, an access domain sub-network and a core domain sub-network. Each sub-network contains a number of items of equipment as its sub-networks. A view of the connection model for the network is illustrated in FIG. 5, and it will be appreciated that this network could itself be a sub-network of the larger network illustrated in FIG. 1.

The cost model of each network can be represented using a cost graph or a set of cost graphs. A cost graph contains the following three basic elements, as follows:

"Cost node" - the basic element in a cost graph. Each cost node has a name and the following information associated with the cost node:

- A set of features this cost node supports, derived from the connection model
- The cost of using these features
- The delay in implementing these features.

"Termination" – a special node in the cost graph indicating the potential starting and ending point of a path. It could be a single termination, although typically it is a termination group.

"Edge" – a line between a termination point and a cost node or between two cost nodes.

An important aspect of constructing a cost graph is to identify a set of cost nodes for the network. Potentially, any network resource or abstract of such resource can be a cost node. Example network resources include network equipment, connections, sub-networks, and even the network itself. Manually building a cost graph to represent a cost model of an entire network is a complex task. Therefore the connection manager allows the cost model of a network to be an aggregation of the cost models of the sub-networks. Hence, the cost model of a network can be composed from a set of cost models of sub-networks or even a cost model of a group of network equipment, with negligible configuration effort. In the example network

shown in FIG. 4, a simple cost model can be constructed for each piece of network equipment, such as access devices and switches.

FIG. 6 shows the multiplexer A1 and its corresponding cost graph. On the customer side of A1, there two termination points, A and B. Both A and B can be grouped as a termination group. The access device does not support local switching, so there are is no direct connection between A and B. On the network side, A1 has a termination a connected to an edge switch in the core domain. Multiplexer A1 can be modeled as a single cost node with two terminations, TG1 (termination group 1 containing terminations A and B) and a. Turning to consider a mulitplexer with reference to FIG. 7. In general, an m-to-n multiplexer has one customer side containing m terminations and one core side containing n terminations. The terminations on each side can be grouped into one group. Such a multiplexer can be represented as a single cost node with two termination groups as shown in FIG. 7B.

The cost model for an edge switch and a core switch is similar to the one for an access device, except that both the edge switch and the core switch support local switching. That is a path can be created from one termination point to itself (it may in fact go through different virtual channels or virtual paths). In the case of an edge switch E1 as shown in FIG. 8A, paths m-E1-m, n-E1-n, m-E1-n, m-E1-q and n-E1-q can be created. To represent this, double edges to the same termination should be used. The cost graph of the edge switch E1 can be represented as shown in FIG. 8B.

Because the physical capabilities of a core switch are similar to that of the edge switch E1, the cost model of a core switch should be similar to the cost model of the edge switch E1. For example, core switch C1 is capable of performing local switching. To represent this, duplicated terminations should be used as shown in FIG. 9. However, a business rule may specify that based on the network architecture given in Figure 1, the local switching capability supported by core switches does not add any value to the path creation, and hence should be ignored during cost modelling. For example, because edge switch E1 supports local switching,

a path A1-E1-A2 can be created. This basically eliminates the requirement of having a path A1-E1-C1-E1-A1. To enforce this business rule, the cost model of core switch C1 should really be modelled shown in FIG. 10. This is an example of how, in the cost model, a business rule can override physical capability in the network.

Cost Model Aggregation

Generally a physical network contains a number of sub-networks and ~~for mapped cost models, the connection manager performs the aggregation.~~ In some embodiments, the cost model may be partially mapped as required. If each sub-network's cost model contains only a single cost graph and a link between a termination of a cost graph to another termination of another cost graph is specified by a system integrator, then connection manager does the aggregation by replacing the terminations and associated edges with a single edge. For example, if a link is specified between the termination q of cost graph E1 and the termination r of cost graph C1, as shown in FIG. 11. The terminations involved in the link q and r become internal terminations (sometimes also referred to as intermediate terminations) of the current network. These terminations will be required by the creation of sub-paths during routing.

If the connections between different sub-networks bear significant cost, then a cost can be specified for the links. In the above example, if the connection between E1 and C1 bears a cost, a new cost node CN_L1 is introduced when aggregating these two cost models, as shown in FIG. 12. If a link involves duplicated terminations (e.g. in the case of support local switching), then the other cost graph (the one without duplicated terminations) should be duplicated and each linked to one of the duplicated terminations. For example, a link between termination a of cost graph for A1 (see Figure 6B) and termination m in cost graph for E1 (see Figure 8B) will lead to the aggregated cost graph illustrated by FIG. 13.

Applying these aggregation techniques to all cost models in the core domain, the domain cost model shown in FIG. 14 can be obtained. Whilst FIG. 15 illustrates the cost model for the network obtained, after accounting

for duplicated terminations, for the cross domain corresponding to the physical network shown in FIG. 4.

Cost Based Route Selection

The method for route selection involves creating an aggregate cost graph of the relevant sub-networks, as described above, and keeping a record of which sub-network is responsible for each cost node. The second part of the method involves finding the lowest-cost of path through the cost nodes between terminations available at the desired locations. A system for allowing exclusion of certain types of sub-networks may, in some cases, be used to "filter" the cost model prior to using the route selection method of the embodiment.

A routing method which may be used to select a route with the lowest cost will now be described. During the selection process, the feature set associated with each cost node provides another level of filtering. If the required features are not supported by a sub-network, then any paths involving that sub-network are not selected. In the following discussion, in order to describe the routing method clearly, a cost graph is re-shaped according to the starting location (or termination group) specified in a path creation request. For example, to create a cross domain path from termination point E to H as indicated in Figure 4, the cross domain cost graph in FIG. 15 can be re-shaped as a tree structure with the starting termination group as the root, and the ending termination group and other terminations as the leaves. The re-shaped cost graph is shown in FIG. 16. Such shaped cost graph shows all the possible routes starting with the given termination. If multiple routes from the starting termination to the ending termination exist, then the ending termination will appear more than once.

The diagram in FIG. 17 illustrates the basic principle of the path selection method. Starting from C1, which is the cost node directly linked to the starting termination A, a first wave is generated. The wave contains all the cost nodes that support the required feature and are directly connected to C1. Three cost nodes involved in the frontage of the first wave are C2, C3 and C4. From the starting termination A to each frontage cost

node forms a candidate route. By adding the current sub-total cost of each candidate route, a lowest-cost candidate route (from A to C2) is selected. Progress will only be made with the currently selected route by pushing the wave one step further to form a second wave. The frontage cost nodes of the second wave include two groups:

- all frontage cost nodes of the unselected routes in the last wave, e.g. C3 and C4;
- all cost nodes directly connected to the selected cost node in last wave, e.g. C5 and C6.

Here it is assumed that both C3 and C4 support the required feature.

The above process can be repeated until the paths from A to B with the lowest cost are selected. In the example the selected route is A-C1-C2-C6-C10-H. The cost of the selected route is 7, which cost is lower than the current sub-total of any other single candidate route. Applying the method for route selection from E to H in FIG. 16, a route from E-CN_A3-CN_E2-CN_E3-CN_A4-H will be selected as follows with reference to FIG. 18. . Apart from the feature parameter, that may affect the route selection, another parameter "delay" may also affect the selection of a route. The routing algorithm can also be optionally arranged to reject paths having delays greater than a required maximum delay.

Once a route is selected using the cost model, it provides a reference of how the required connection can be created in the physical network. If the selected route only involves one sub-network the connection can be created over the sub-network. If the route involves more than one sub-network, then a number of paths need to be created over the relevant sub-networks. In order to make the necessary connections, the current network must find the intermediate terminations at the boundary of each sub-network. These terminations are those used to form links, as discussed above in relation to FIG. 11.

It will be appreciated that implementations of the cost model, other than a selection method that uses static data in nodes on a traversible

graph as described above in relation to the preferred embodiment, are possible. In general terms, the selection method of the embodiment is one implementation of a cost model wherein a delegate offers a data structure that is capable of representing complex or subtle cost predictions when interpreted by a method pre-agreed by the delegate and delegator. Though such data structures are complex, they are not general purpose, in the sense of being Turing machines. This means that there will always be some cost predications that cannot be conveniently represented in the model.

An alternate approach is to pass a piece of data that is the program for a Turing machine. The delegator and delegate must still agree of the semantic of the data (that is, the implementation of the Turing machine). However there will no longer be intrinsic limitations of the ability to represent any cost model. For example, it is unlikely that a data structure approach could model a cost prediction that involves the calculation of B-spline interpolation (unless the data structure designer foresaw that need). The Turing machine approach does not suffer such a limitation. A practical implementation of the Turing machine approach is to agree on an implementation that is well understood in the industry. One such example is to use a Java Virtual Machine implementation, and the cost model is then transferred as a sequence of Java byte codes.

The cost model allows the connection manager to expose or publish a comprehensive estimate of what a path between two locations will cost, without the client having to make a vast number of queries about the cost of each possible choice. It is the service provider's or network owner's choice as to how comprehensive a cost model is provided. They may publish a detailed cost model without exposing the underlying structure of the network.

By automating the routing and configuration of connections across complex networks, the connection manager substantially reduces the need for manual management at the network and element levels. Connections can be provisioned in real time and the connection manager will scale to process increasing volumes of new connections as broadband communications networks grow.

The object oriented approach to modeling aspects of the network in the connection manager, wherein the abstract connection model is differentiated from the connection model instantiations result in a high degree of reuse of clients and servers. This approach also allows for, but does not enforce, very flexible client and server implementations, which can match a rapidly changing business scenario.

Throughout the specification the aim has been to describe the preferred embodiments of the invention without limiting the invention to any one embodiment or specific collection of features.

Dated this Twelfth day of October 1998

CiTR Pty Ltd

by its Patent Attorneys

Fisher Adams Kelly

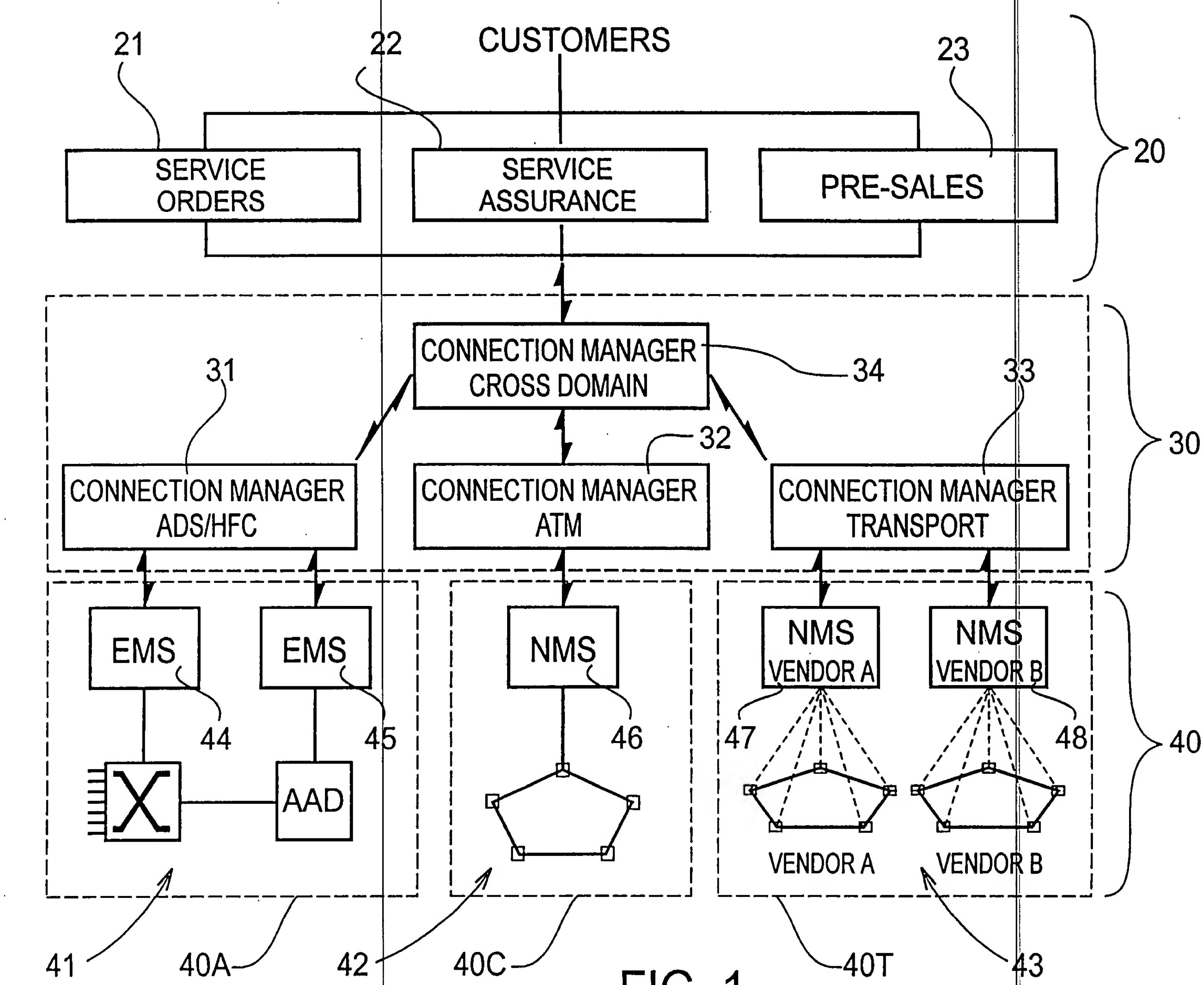

CUSTOMERS
21
22
23
SERVICE ORDERS
SERVICE ASSURANCE
PRE-SALES
20
CONNECTION MANAGER CROSS DOMAIN
34
31
33
32
30
CONNECTION MANAGER ADS/HFC
CONNECTION MANAGER ATM
CONNECTION MANAGER TRANSPORT
EMS
EMS
NMS
NMS VENDOR A
NMS VENDOR B
44
45
46
47
48
40
AAD
VENDOR A
VENDOR B
41
40A
42
40C
40T
43

FIG. 1

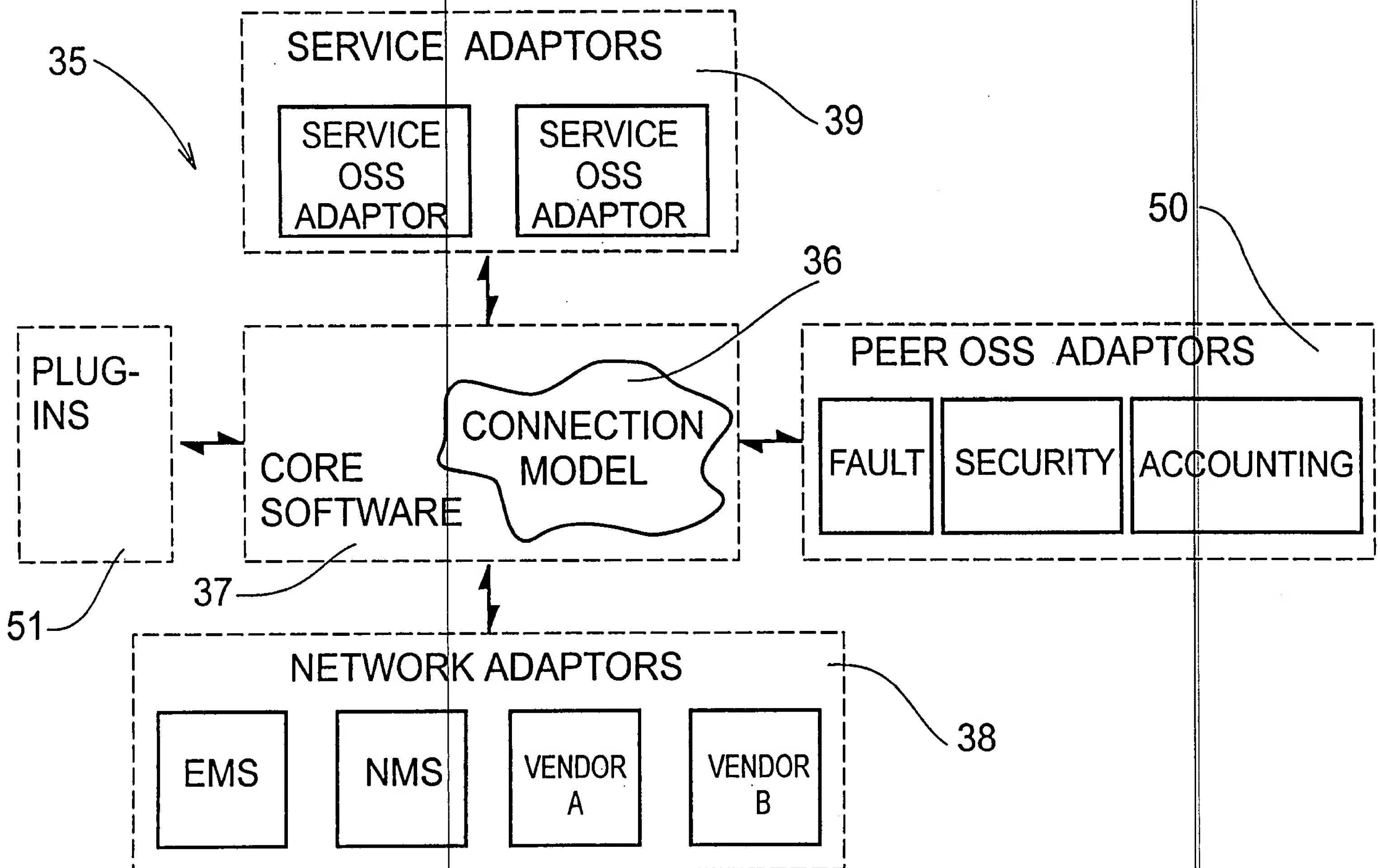

35
SERVICE ADAPTORS
SERVICE OSS ADAPTOR
SERVICE OSS ADAPTOR
39
36
50
PLUG-INS
CORE SOFTWARE
CONNECTION MODEL
PEER OSS ADAPTORS
FAULT
SECURITY
ACCOUNTING
37
51
NETWORK ADAPTORS
EMS
NMS
VENDOR A
VENDOR B
38

FIG. 2

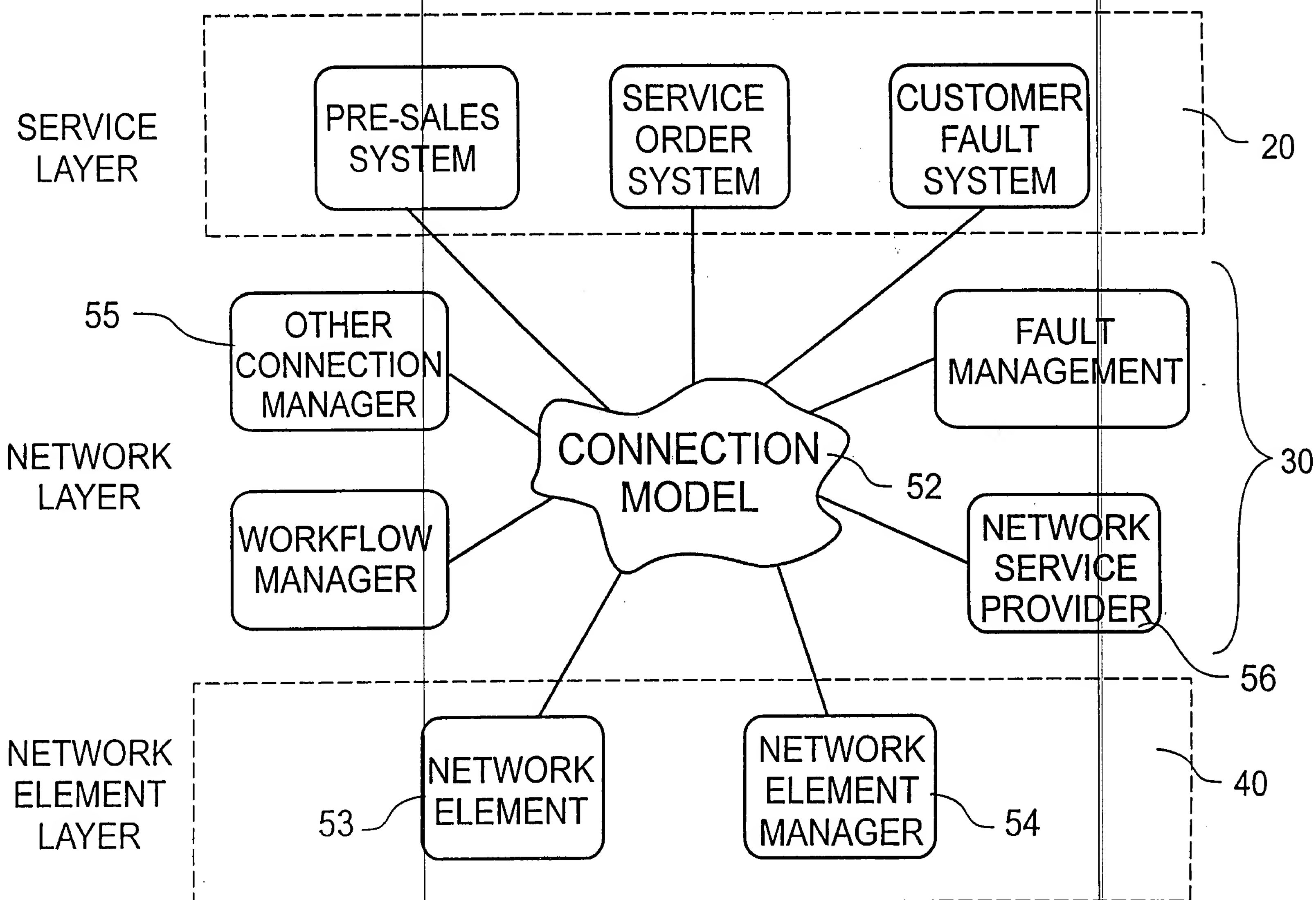

SERVICE LAYER
PRE-SALES SYSTEM
SERVICE ORDER SYSTEM
CUSTOMER FAULT SYSTEM
20
55
OTHER CONNECTION MANAGER
NETWORK LAYER
WORKFLOW MANAGER
CONNECTION MODEL
52
FAULT MANAGEMENT
30
NETWORK SERVICE PROVIDER
56
NETWORK ELEMENT LAYER
53
NETWORK ELEMENT
NETWORK ELEMENT MANAGER
54
40

FIG. 3

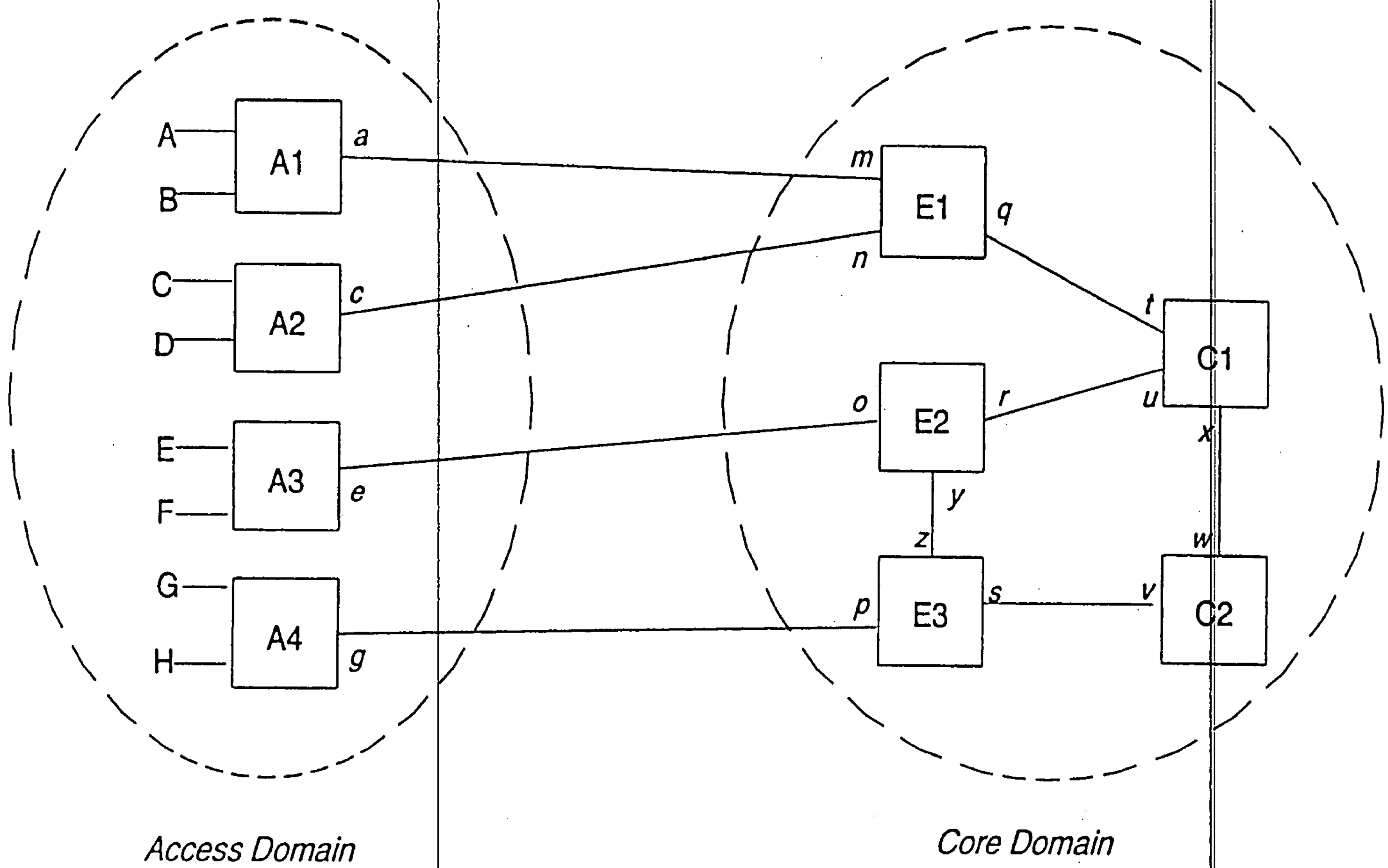
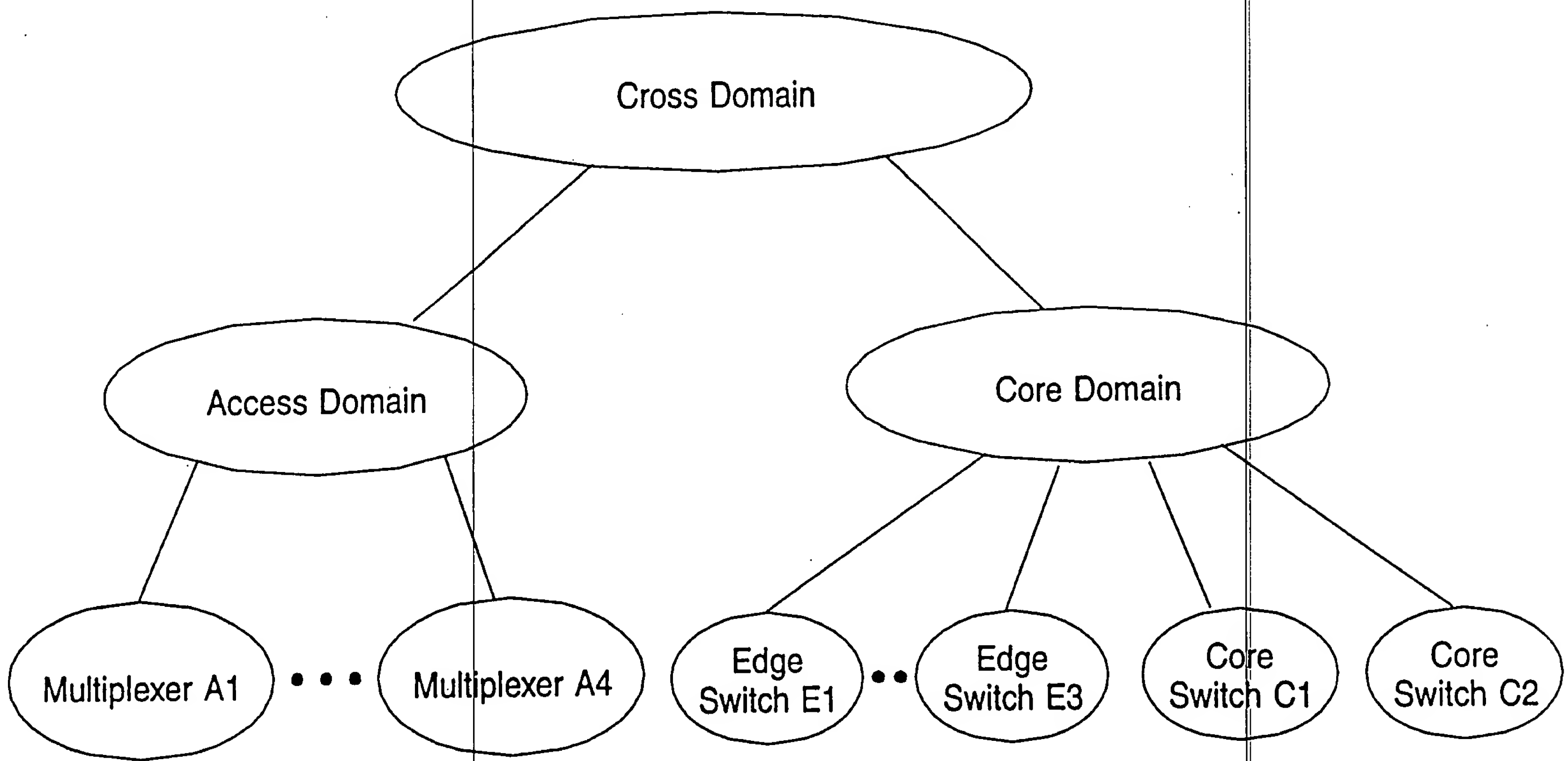

A
B
A1
a
C
D
A2
c
E
F
A3
e
G
H
A4
g
m
n
E1
q
o
E2
r
y
z
p
E3
s
t
u
C1
x
w
v
C2
Access Domain
Core Domain

FIG. 4



Cross Domain
Access Domain
Core Domain
Multiplexer A1
Multiplexer A4
Edge Switch E1
Edge Switch E3
Core Switch C1
Core Switch C2

FIG. 5

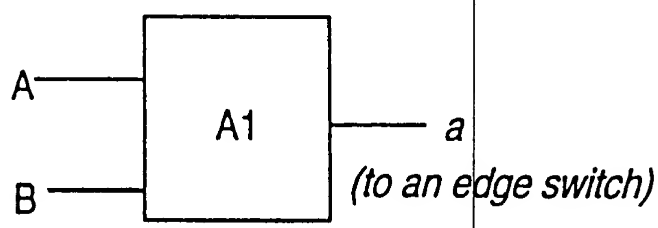
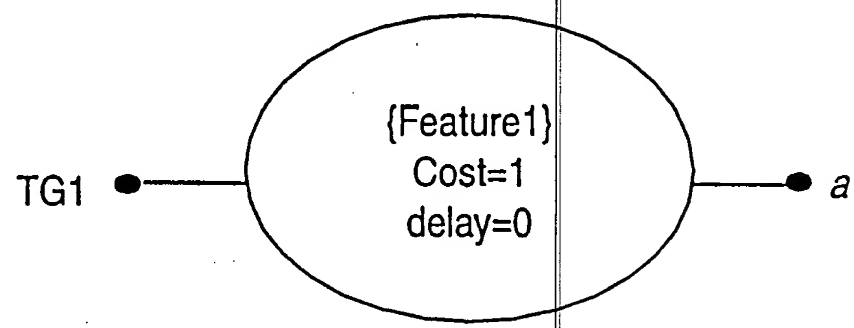


FIG. 6A



CostNode CN_A1

FIG. 6B

Physical View of an m-to-n Access Device

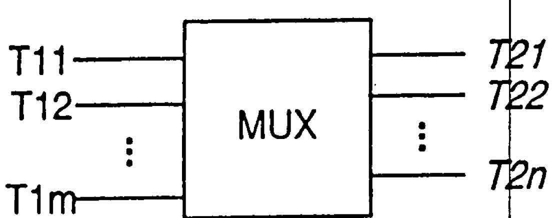
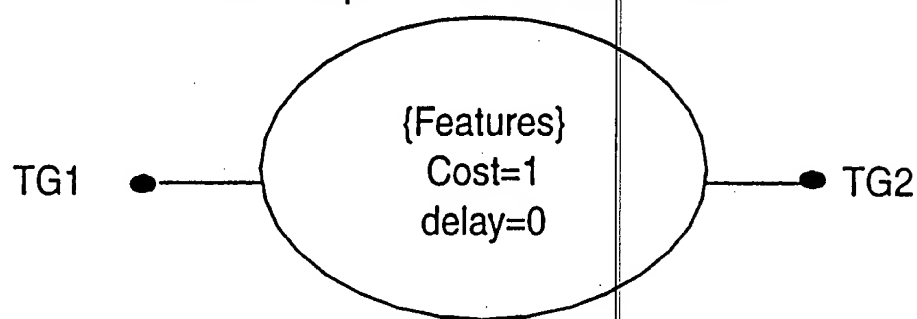


FIG. 7A

Cost Graph of the Access Device



CostNode CN_MUX

FIG. 7B

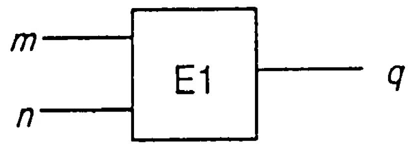


FIG. 8A

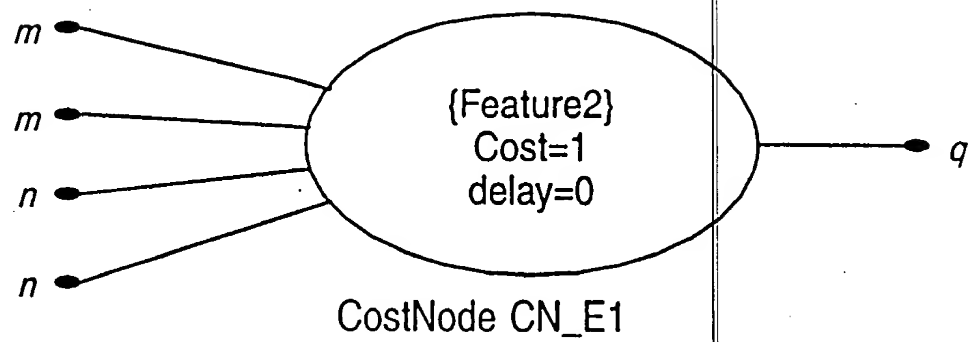


FIG. 8B

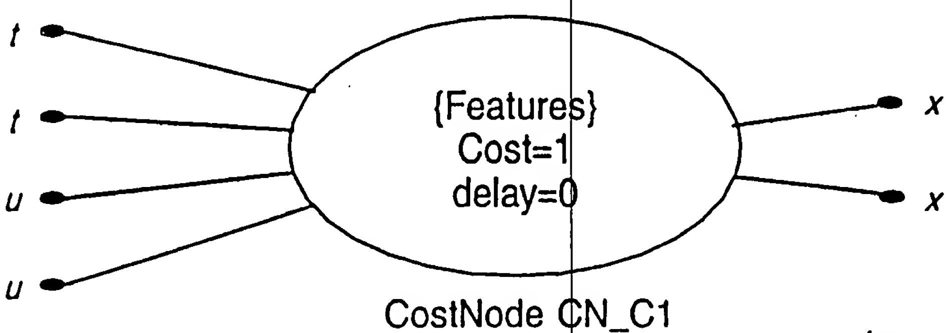


FIG. 9

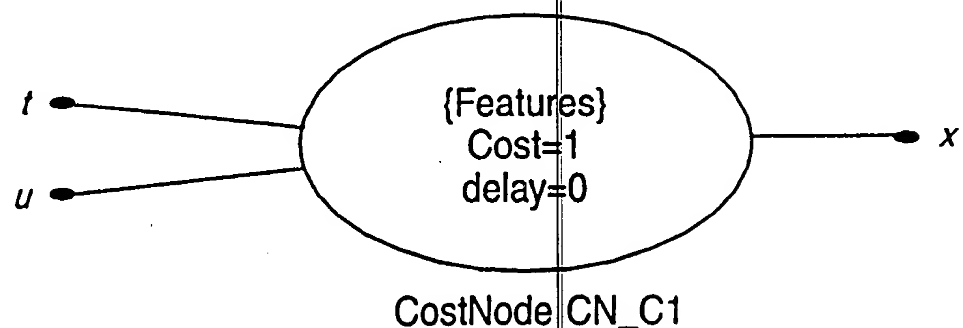


FIG. 10

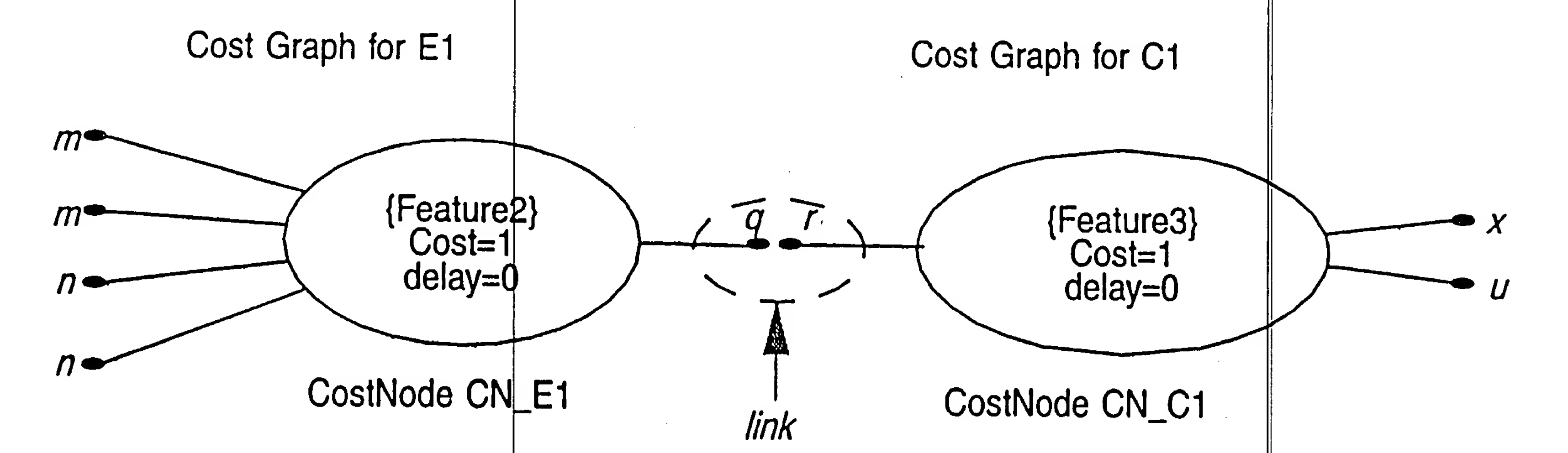

Cost Graph for E1
Cost Graph for C1
m
m
n
n
{Feature2}
Cost=1
delay=0
q
r
{Feature3}
Cost=1
delay=0
x
u
CostNode CN_E1
link
CostNode CN_C1

FIG. 11

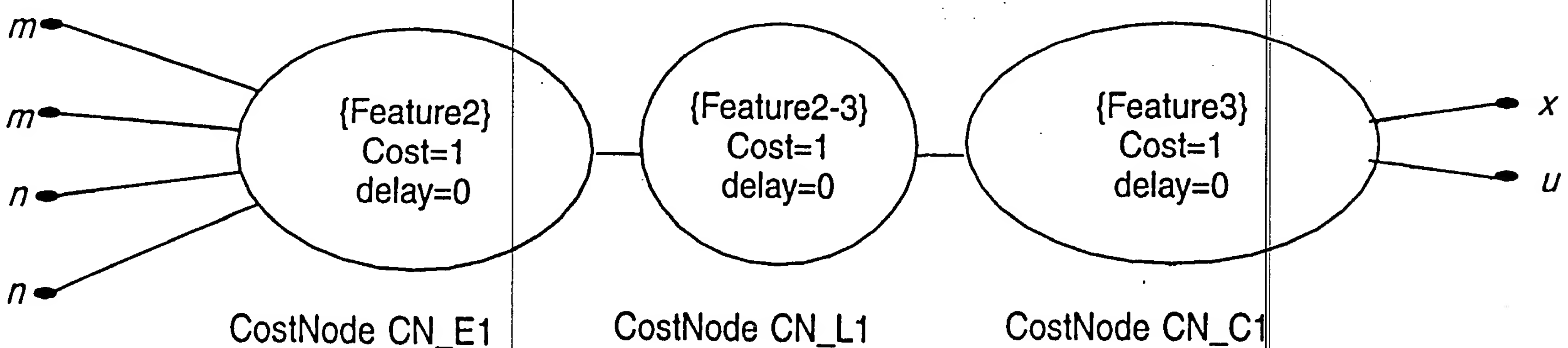

m
m
n
n
{Feature2}
Cost=1
delay=0
{Feature2-3}
Cost=1
delay=0
{Feature3}
Cost=1
delay=0
x
u
CostNode CN_E1
CostNode CN_L1
CostNode CN_C1

FIG. 12

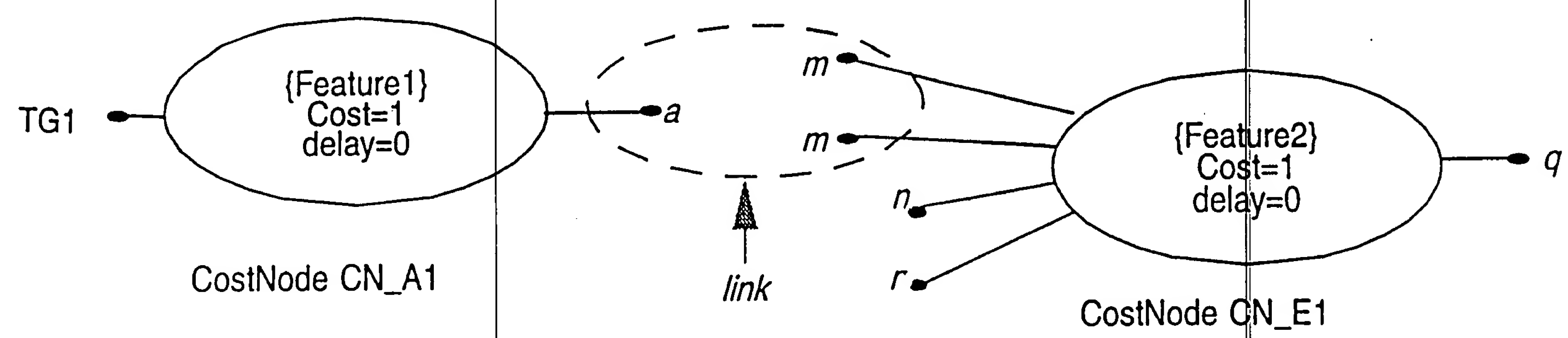


FIG. 13A

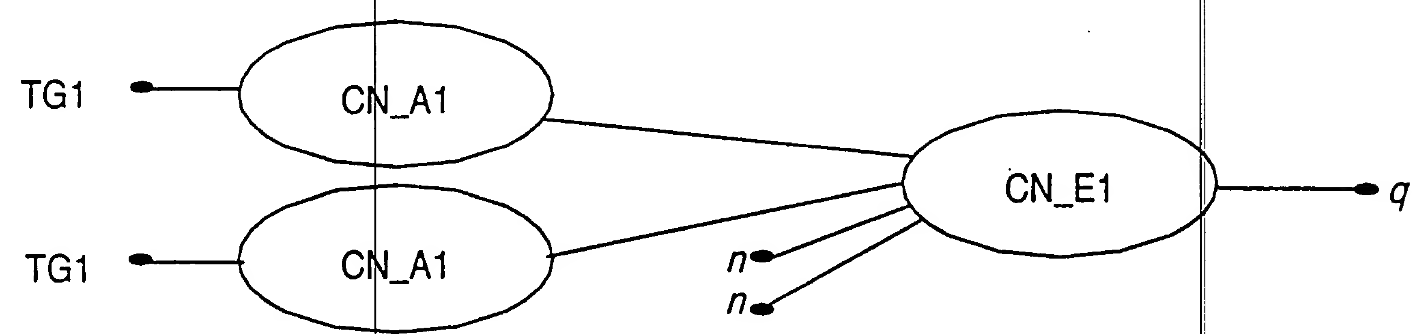


FIG. 13B

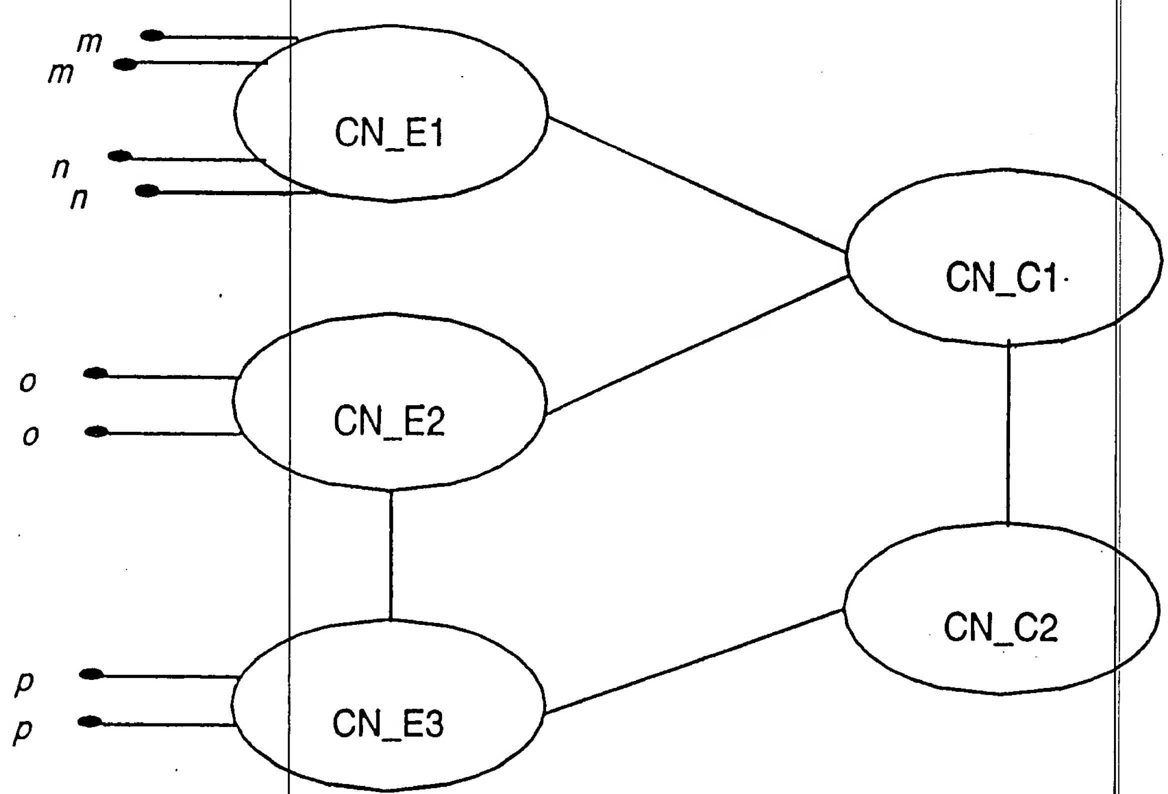


FIG. 14

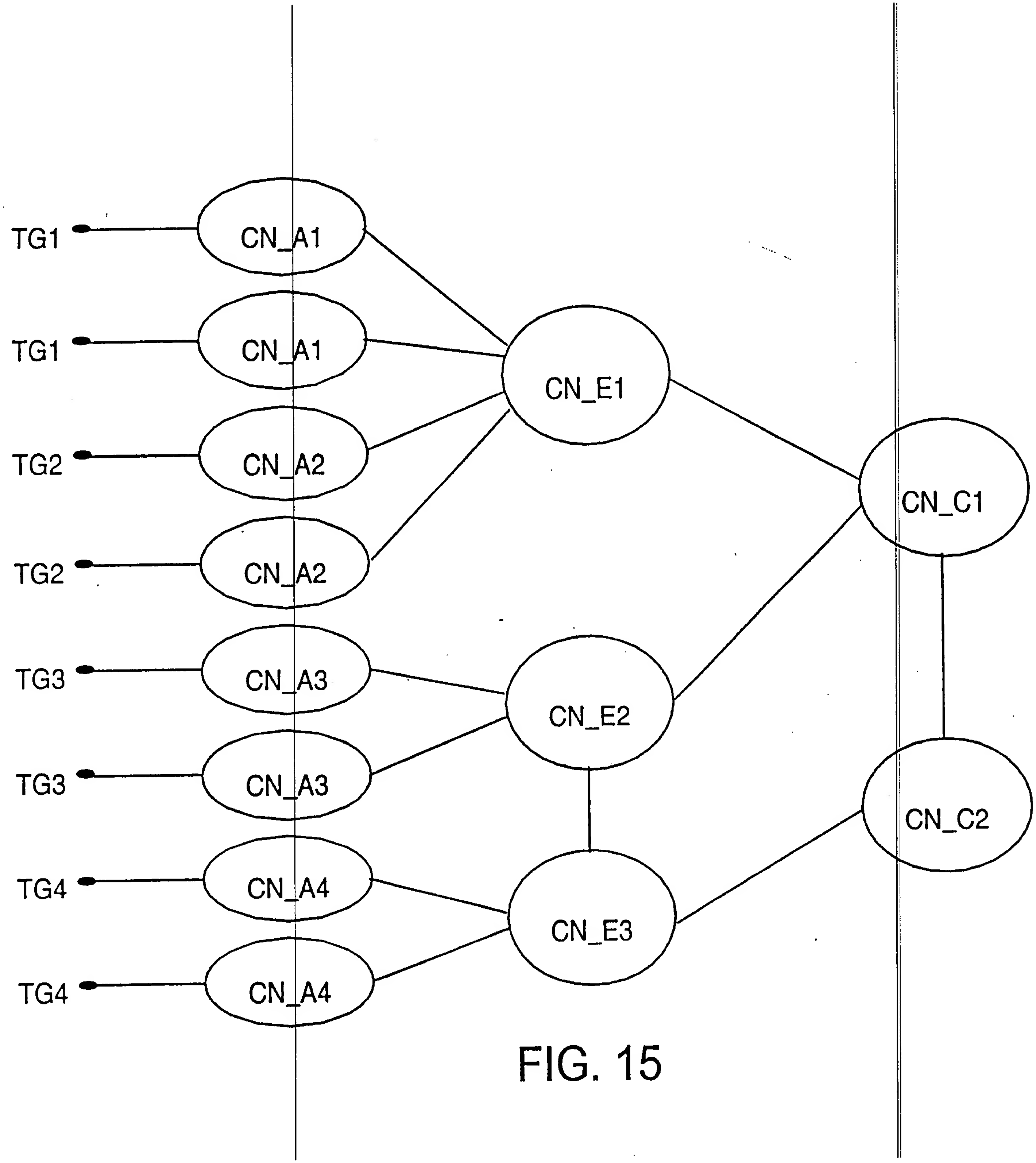
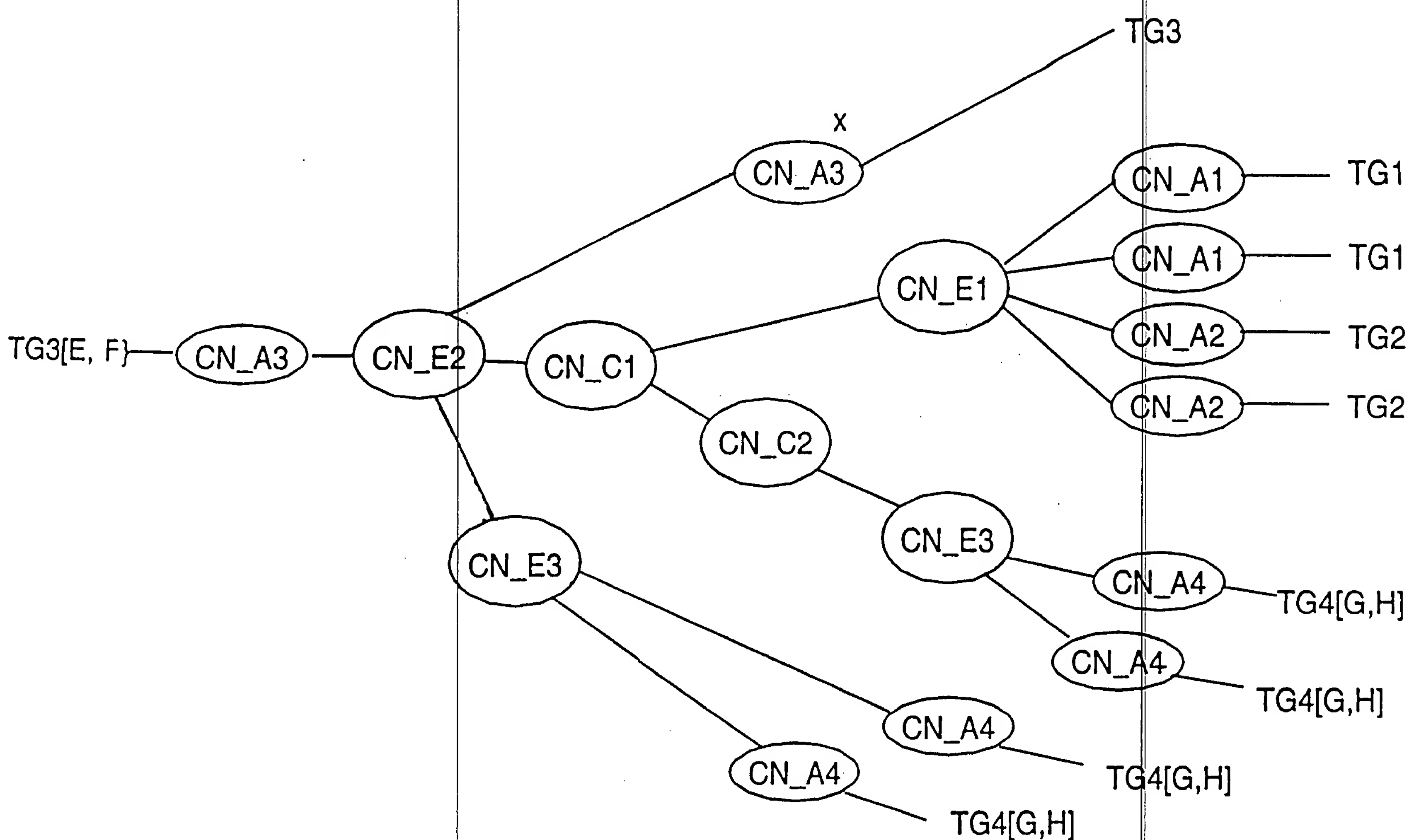
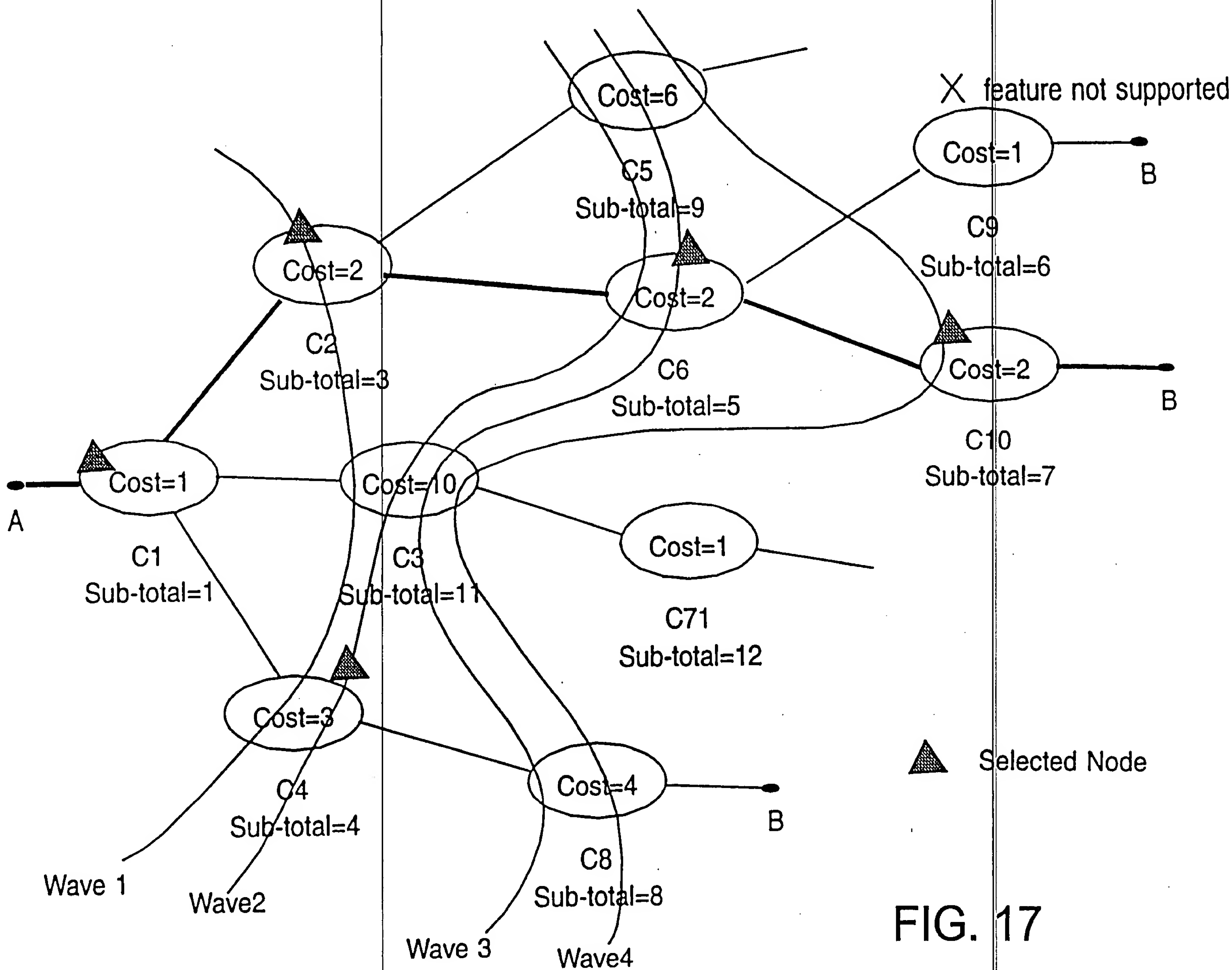

TG1
CN_A1
TG1
CN_A1
TG2
CN_A2
TG2
CN_A2
TG3
CN_A3
TG3
CN_A3
TG4
CN_A4
TG4
CN_A4
CN_E1
CN_E2
CN_E3
CN_C1
CN_C2

FIG. 15



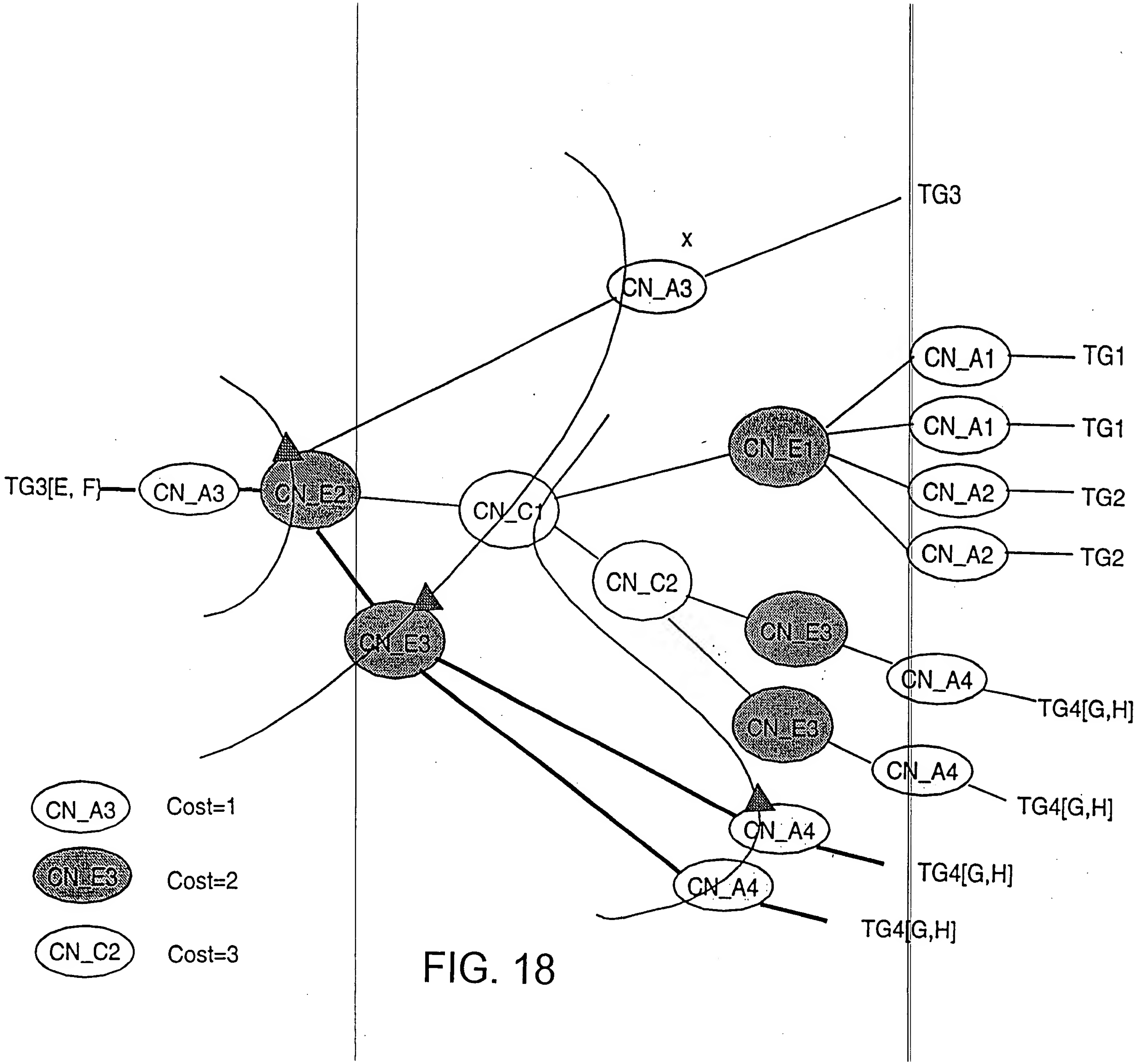
TG3
x
CN_A3
CN_A1
TG1
CN_A1
TG1
CN_E1
CN_A2
TG2
CN_A2
TG2
TG3[E, F}
CN_A3
CN_E2
CN_C1
CN_C2
CN_E3
CN_E3
CN_A4
TG4[G,H]
CN_A4
TG4[G,H]
CN_A4
TG4[G,H]
CN_A4
TG4[G,H]

FIG. 16



X feature not supported
Cost=6
Cost=1
B
C5
Sub-total=9
C9
Sub-total=6
Cost=2
Cost=2
C2
Sub-total=3
C6
Sub-total=5
Cost=2
B
C10
Sub-total=7
Cost=1
Cost=10
A
C1
Sub-total=1
C3
Sub-total=11
Cost=1
C71
Sub-total=12
Cost=3
Cost=4
B
Selected Node
C4
Sub-total=4
C8
Sub-total=8
Wave 1
Wave2
Wave 3
Wave4

FIG. 17



TG3
x
CN_A3
CN_A1
TG1
CN_A1
TG1
CN_E1
CN_A2
TG2
CN_A2
TG2
TG3[E, F]
CN_A3
CN_E2
CN_C1
CN_C2
CN_E3
CN_E3
CN_A4
TG4[G,H]
CN_A4
TG4[G,H]
CN_E3
CN_A4
TG4[G,H]
CN_A4
TG4[G,H]
CN_A3
Cost=1
CN_E3
Cost=2
CN_C2
Cost=3

FIG. 18